

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

AUTOMATICKÁ EVALUACE MULTI-BIOMETRICKÝCH SYSTÉMŮ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

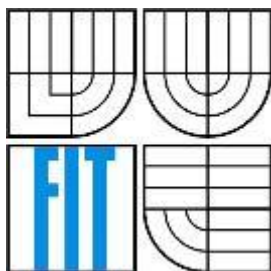
RICHARD ŠTUPÁK

BRNO

2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

AUTOMATICKÁ EVALUACE MULTI-BIOMETRICKÝCH SYSTÉMŮ

AUTOMATIC EVALUATION OF MULTI-BIOMETRIC SYSTEMS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RICHARD ŠTUPÁK

VEDOUCÍ PRÁCE

SUPERVISOR

ING ŠTĚPÁN MRÁČEK

BRNO

2014

Abstrakt

Táto bakalárska práca sa zaoberá problematikou evaluácie multi-biometrických systémov a spôsobmi fúzie biometrických vlastností. Prvá časť vysvetľuje pojmy z biometrie, biometrických a multi-biometrických systémov. V ďalšej časti je popis návrhu a implementácie jednotlivých častí aplikácie. Posledná časť obsahuje experimenty s vytvorenou aplikáciou na testovacích dátach. Cieľom testovania bolo overiť, či aplikácia dokáže nájsť spôsob fúzie za dosiahnutím lepších výsledkov.

Abstract

The objective of the bachelor thesis is evaluation of multi-biometric systems and types of fusion on biometric data. First part explains terms from biometric and multi-biometric systems. Next part describes concept and implementation of individual parts of application. Last part contains experiments with developed application on test data. Purpose of testing was to confirm, that application can find type of fusion with better results.

Klíčová slova

Biometria, multi-biometrický systém, fúzia, pravdepodobnosť chybného odmietnutia, pravdepodobnosť chybného prijatia,

Keywords

Biometric, multi-biometric, score-level fusion, feature-level fusion, decision-level fusion, false rejection rate, false recognition rate

Citace

Štupák Richard: Automatická evaluace multi-biometrických systémů, Brno, FIT VUT v Brně, 2014

Automatická evaluace multi-biometrických systémů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Štěpana Mráčka.

Další informace mi poskytli...

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Richard Štupák

20.5.2014

Poděkování

Týmto by som chcel poďakovať vedúcemu mojej práce, Ing. Štěpanovi Mráčkovi, za jeho ochotu a venovaný čas a tiež za odborné rady.

© Richard Štupák, 2014

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

| | |
|---|----|
| Obsah..... | 1 |
| 1 Úvod..... | 3 |
| 2 Biometrické systémy..... | 4 |
| 2.1 Úvod do biometrie | 4 |
| 2.2 Všeobecné princípy biometrických systémov | 5 |
| 2.2.1 Zber dát..... | 5 |
| 2.2.2 Prenos dát..... | 6 |
| 2.2.3 Spracovanie signálu | 6 |
| 2.2.4 Proces rozhodovania..... | 7 |
| 2.2.5 Uloženie dát | 7 |
| 2.3 Verifikácia a identifikácia..... | 7 |
| 2.4 Meranie výkonnosti biometrických metód a zariadení..... | 8 |
| 2.4.1 Pravdepodobnosť chybného odmietnutia (False Rejection Rate – FRR) | 8 |
| 2.4.2 Pravdepodobnosť chybného prijatia (False Acceptance Rate – FAR)..... | 9 |
| 2.4.3 Vzťah FRR a FAR | 10 |
| 2.5 Multi-biometrické systémy | 11 |
| 2.5.1 Úvod do multi-biometrických systémov..... | 11 |
| 2.5.2 Taxonómia multi-biometrických systémov | 11 |
| 2.5.3 Úrovne fúzie | 12 |
| 3 Návrh riešenia a implementácia | 14 |
| 3.1 Načítanie dát | 14 |
| 3.2 Porovnanie a metriky | 15 |
| 3.3 EER – Equal Error Rate..... | 16 |
| 3.4 Práca s vektormi | 17 |
| 3.5 Normalizácia..... | 17 |
| 3.6 Fúzia na úrovni vlastností (<i>feature-level</i>) | 18 |
| 3.7 Fúzia na úrovni skóre (<i>score-level</i>) | 20 |
| 3.8 Fúzia na úrovni rozhodovania (<i>decision-level</i>)..... | 21 |
| 3.9 Výsledná aplikácia..... | 22 |
| 4 Experimenty | 23 |
| 4.1 Biometrická databáza..... | 23 |
| 4.2 Použitý HW | 23 |
| 4.3 Fúzia na úrovni vlastností..... | 23 |
| 4.4 Fúzia na úrovni skóre..... | 27 |

| | | |
|-----|-----------------------------------|----|
| 4.5 | Fúzia na úrovni rozhodnutia | 30 |
| 4.6 | Zhodnotenie | 32 |
| 5 | Záver | 33 |

1 Úvod

Pred necelými štyridsiatimi rokmi, IBM navrhlo aby užívateľ počítača mohol byť rozpoznaný na počítačovom terminály „Na základe niečoho, čo vie alebo čo si pamätá... Na základe nejakého predmetu, ktorý vlastní... Na základe fyzickej charakteristiky“. Táto analýza bola prevádzaná v kontexte s dátovou bezpečnosťou – vzdialene rozpoznať tých, ktorí boli autorizovaný pre prístup k uloženým dátam. Od vtedy, automatické rozpoznanie osoby na základe fyzických vlastností a charakteristík správania, bolo známe ako „biometrická autentizácia“ a jej použitie bolo rozšírené do rôznych ďalších oblastí.

Táto práca sa zaoberá rozpoznávaním osôb na základe kombinácii biometrických vlastností. Biometria, je veda zaoberajúca sa rozpoznávaním identity osôb na základe fyzických a behaviorálnych atribútov osoby, ako sú napríklad tvár, odtlačky prstov, hlas a dúhovka. Bakalárska práca je zameraná na problematiku fúzie jednotlivých biometrických vlastností do fungujúceho celku. V prvej časti sú vysvetlené pojmy súvisiace s biometriou a biometrickými systémami. V druhej časti je popísaný návrh a spôsob implementácie jednotlivých častí aplikácie. Účelom aplikácie je evaluácia zadaného multi-biometrického systému a zistenie spôsobu fúzie pre dosiahnutie lepšieho výsledku. Posledná časť je venovaná experimentovaniu s vytvorenou aplikáciou a porovnaním výsledkov.

2 Biometrické systémy

V nasledujúcej časti budú vysvetlené základné veci o biometrii a biometrických systémoch. Uvedieme základný princíp fungovania biometrických systémov, princíp verifikácie a identifikácie. Povieme si niečo o meraní výkonnosti biometrických systémov a načrtujeme základy multi-biometrických systémov.

Výklad pojmov a vzorce, ktoré sú uvedené v tejto kapitole boli čerpané z kníh *Handbook of biometrics* [1][2] a *Biometria a identita človeka ve forenzních a komerčních aplikacích* [3]. Autormi prvej knihy sú *Anil K. Jain, Patrick Flynn a Arun A. Ross*. Jedná sa o knihu v anglickom jazyku, ktorá popisuje základy biometrických a mutli-biometrických systémov a viac sa zaoberá jednotlivými druhmi biometrických vlastností. Autormi druhej knihy sú *Roman Rak, Václav Matyáš, Zděnek Říha a kolektiv*. Táto kniha popisuje základy biometrie, jednotlivé druhy biometrických vlastností a ich využitie v praxi.

2.1 Úvod do biometrie

V dnešnej dobe kvôli obavám z bezpečnostných rizík zvažujú súkromné spoločnosti, banky, poisťovne a vládne inštitúcie čo najkvalitnejšie systémy zabezpečenia prístupu do svojich priestorov a sietí obsahujúcich citlivé dáta. Využíva sa rozpoznanie na základe hsiel, identifikačných kariet a najmodernejší spôsob overovania totožnosti je biometria.

Biometria je veda, ktorá sa zaoberá rozpoznávaním identity ľudí na základe fyzických, chemických alebo behaviorálnych atribútov osoby. Vhodnosť biometrie v modernej spoločnosti bola posilnená potrebou identifikačných systémov veľkého rozsahu využívaných v rôznych kontextoch. Využitie týchto systémov zahŕňa napríklad zdieľanie zdrojov sieťových počítačov, povolenie prístupu k jadrovým zariadeniam alebo vykonávanie finančných transakcií. Šírenie webovo-založených služieb (napr., internet banking) ešte viac podtrhli potrebu spoľahlivých identifikačných systémov, ktoré dokážu spracovávať veľké množstvo osôb.

Najťažšou úlohou identifikačných systémov je rozhodnutie (alebo overenie) identity osoby.¹ Rozpoznávanie identity síce môže byť dôležitá z rôznych dôvodov ale hlavným zmyslom identifikačných systémov je zabránenie prístupu neoprávnených užívateľov k chráneným zdrojom. Tradičné metódy, ktoré sa využívajú pri stanovení identity osoby zahŕňujú mechanizmy, ktoré využívajú nejakú vedomosť alebo vec, ktorú vie resp. vlastní daná osoba. Avšak tieto veci reprezentujúce identitu osoby sa dajú ľahko stratiť, požičať, zmanipulovať alebo ukradnúť a tým bude

¹ *Identita osoby* - môže byť videná ako informácie spojené s danou osobou v danom identifikačnom systéme. Napríklad, bankou vydávané kreditné karty sú typický spojené so zákazníkovým menom, heslom, číslom občianskeho preukazu, adresou a dátumom narodenia. Takže identita zákazníka v tomto systéme bude definovaná týmito atribútmi (meno, adresa, a i.)

kompromitovaná bezpečnosť systému. Biometria ponúka prirodzené a spoľahlivé riešenie určitých aspektov identifikačných systémov použitím automatických alebo poloautomatických zariadení pre rozpoznanie osoby na základe jej biologických charakteristík. Použitím biometrie je možné určiť identitu na základe toho *kto sme*, miesto toho *čo vlastníme* alebo *čo si pamätáme*.

Efektivita autentifikátora je založená na základe jeho vhodnosti pre daný systém a tiež na základe robustnosti voči rôznym typom útokov. O’Gorman uvádza zoznam rôznych útokov, ktoré môžu byť použité na systémy založené na hesle alebo *odznaku*: (1) útok na klienta (napr. hádanie hesla, ukradnutie odznaku). (2) útok na poskytovateľa (napr. získanie súboru s heslami). (3) odpočúvanie (napr. nazeranie pri zadávaní hesla). (4) zavrhnutie (napr. tvrdenie, že karta bola odcudzená). (5) útok trojským koňom (napr. inštalácia softwaru na získavanie logu z obrazovky). (6) DoS² útok. Niektoré z týchto útokov sa dajú odraziť zahrnutím určitých obranných mechanizmov, ale nie je možné zvládnuť všetky problémy spojené s používaním hesiel a *odznakov*.

Biometrické systémy využívajú rôzne druhy fyzických a behaviorálnych charakteristík, napr. tváre, geometriu ruky alebo prsta, dúhovku, podpis, chôdzu, odtlačok dlane, hlasový záznam, pach alebo DNA informáciu osoby, na určenie identity. Napriek určitým limitom biometrických systémov, majú tieto systémy navrch oproti tradičným bezpečnostným mechanikám, pretože biometrické vlastnosti sa nedajú ľahko ukradnúť alebo zdieľať. Okrem vylepšenia bezpečnosti, biometrické systémy sú taktiež pre užívateľa pohodlnejšie, keďže si nemusí vymýšľať a pamätať heslá.

2.2 Všeobecné princípy biometrických systémov

Každé biometrické spracovanie má päť základných etáp:

1. Zber (snímanie) biometrických dát
2. Prenos dát
3. Spracovanie nameraného signálu
4. Proces rozhodovania
5. Uloženie dát

2.2.1 Zber dát

Biometrické spracovanie začína meraním anatomicko-fyziologických alebo behaviorálnych charakteristík človeka, teda zberom (snímaním) biometrických dát. Základným predpokladom identifikácie (verifikácie) je jednoznačnosť identifikačných charakteristík, a ich merateľnosť a časová stálosť (nemennosť). Teda meranie biometrických charakteristík musí byť opakovateľný jav a pri každom meraní musíme dosiahnuť rovnakých nameraných hodnôt, alebo hodnôt s dopredu definovanou a známou odchýlkou.

² *Denial of service* - technika útoku na nejakú službu, s cieľom znefunkčniť túto službu

2.2.2 Prenos dát

Niektoré biometrické systémy, ale nie všetky, zbierajú biometrické dáta na jednom mieste a skladujú ich a/alebo spracovávajú na inom mieste. V týchto prípadoch je nutné zabezpečiť prenos dát. Biometrické systémy pracujú všeobecne s pomerne veľkými objemami dát. Aby prenos dát a ich uloženie bolo efektívne (rýchle, malé nároky na skladovacie priestory, a i.), sú pred prenosom a spracovaním komprimované³. Po ich prenose, pred ďalším spracovaním sa dáta dekomprimujú. Proces komprimácie a dekomprimácie vo všeobecne zapríčiňuje stratu v kvalite dekomprimovaného signálu. Straty sa zväčšujú s rastúcim komprimačným pomerom.

2.2.3 Spracovanie signálu

Spracovanie biometrického signálu môžeme formálne rozdeliť do troch častí

1. Extrakcia unikátnych biometrických charakteristík zo vzorku
2. Kontrola kvality
3. Vyhľadávanie v databáze porovnaním s ďalšími vzorkami šablón

Extrakcia šablóny

Extrakcia prebieha z dekomprimovaného biometrického vzorku a je nezávislá na komprimačných a dekomprimačných algoritmoch. Proces extrakcie je spravidla automatizovaný a jeho cieľom je nájsť a definovať jednoznačné identifikačné charakteristiky. Výsledkom extrakcie je tzv. **šablóna**, pod ktorou si môžeme predstaviť minimálnu množinu (maticu, funkčnú závislosť a i.) bodov, obrazov, objektov a pod., ktoré úplne odrážajú individuálnu špecifickosť preverovaného jedinca pomocou zvolenej biometrickej metódy a teda splňujú základné identifikačné podmienky – unikátnosť, presnosť, časovú nemennosť, technologickosť spracovania. Šablóna je teda matematické vyjadrenie fyzickej (chemickej, biologickej apod.) podstaty biometrického vzorku. Samotné porovnanie a vyhodnotenie biometrických charakteristík pracuje obvykle iba so šablónami.

Kontrola kvality

Po extrakcii identifikačných charakteristík, niekedy dokonca pred ňou alebo behom nej, potrebujeme vedieť, či získaný signál – biometrická vzorka zo snímacieho senzora je dostatočne kvalitný. Keď nasnímané charakteristiky „nedávajú zmysel“, alebo sú akýmkoľvek spôsobom nedostatočné, konštatujeme, že prijatý signál s biometrickou vzorkou je nedostatočný a požadujeme novú, kvalitnú vzorku, pokiaľ je osoba ešte v blízkosti zariadenia pre vstup.

³ *Komprimácia*- technologický, počítačom podporovaný proces, ktorého cieľom je podstatné zníženie objemu komprimovaného súboru. Rozlišujeme dva druhy komprimačných metód a to stratové alebo bezstratové. Opakom komprimácie je *dekomprimácia* – teda návrat do pôvodného stavu

Porovnanie šablón

Extrahované, kvalitné charakteristiky biometrického vzorku – šablóna, ktorá má malé rozmery, je odoslaná do porovnávacieho procesu. Porovnanie prebieha medzi šablónou práve nasnímaného vzorku a skôr nasnímanými šablónami uloženými v databáze (rovnakej osoby alebo iných osôb). Prvé ukladanie tzv. *referenčné šablóny* do databáze sa v praxi nazýva zavedenie (*enrollment*) šablóny (*template*).

Cieľom porovnania šablón je stotožniť načítanú šablónu s jednou alebo viacerými šablónami, uloženými v databázovom systéme. Rozlišujeme dva základné druhy porovnania **identifikácia** a **verifikácia**, rozdiel medzi nimi bude vysvetlený v ďalšej kapitole.

2.2.4 Proces rozhodovania

Účelom porovnania je nájsť dostatočne prijateľnej zhodnosti predkladanej šablóny s referenčnou šablónou, a tým stanovenie alebo potvrdenie identity. Pri porovnávaní sú nájdené vyhovujúce šablóny. Ďalšou etapou spracovania biometrických údajov je rozhodovací proces, v ktorom sa stanovuje identifikačný záver. Na základe merateľných charakteristík, vypracovaných metód a algoritmov, sa stanoví miera zhodnosti medzi predloženou a nájdenou šablónou. V komerčných systémoch je tento proces automatizovaný a aplikácia iba automaticky povie: môžeme/nemôžeme túto osobu autorizovať k ďalšej činnosti.

Oprávnený užívateľ by mal mať zaistené vždy pozitívne vyhodnotenie a nemal by byť odmietaný z dôvodu akejkoľvek technologickej chyby. Tento stav sa nazýva „*falošné odmietnutie oprávneného užívateľa – false rejection*“.

Naopak, neoprávnený užívateľ, musí byť systémom vždy rozpoznaný a odmietnutý. Aj tu však môže nastať chyba, ktorá sa nazýva „*nesprávne prijatie neoprávneného užívateľa – false acceptance*“.

2.2.5 Uloženie dát

V závislosti na type biometrického systému je v databázovom systéme uložená jedna alebo viacero referenčných šablón. V niektorých prípadoch sa ukladajú aj dáta, ktoré sú nasnímané priamo biometrickými senzormi. Referenčné šablóny sú ukladané preto, aby v budúcnosti bolo možné rýchle porovnanie šablóny preverovanej osoby alebo stopy s referenčnými údajmi.

2.3 Verifikácia a identifikácia

V závislosti na kontexte aplikácie môže biometrický systém fungovať v identifikačnom alebo verifikačnom móde.

Verifikácia

Pri verifikácii systém overuje identitu osoby porovnaním nasnímaných biometrických dát s jeho vlastnou biometrickou šablónou uloženou v databáze. V tomto prípade, osoba, ktorá chce byť rozpoznaná, poskytne systému identitu, typicky pomocou PINu, užívateľského mena alebo čipovej karty. Systém následne prevedie porovnanie 1:1 a určí, či poskytnutá identita patrí osobe, ktorej biometrické informácie boli práve nasnímané. Verifikácia sa typicky používa na pozitívne rozpoznávanie, kde cieľom je zabránenie viacerým osobám používať rovnakú identitu.

Identifikácia

Pri identifikácii systém rozpoznáva osobu vyhľadávaním v šablónach všetkých užívateľov. Takže systém prevádza porovnávanie 1:n pre určenie identity osoby. Umožňuje pozitívnu a negatívnu identifikáciu. Cieľom *pozitívnej identifikácie* je zabrániť používania identity jednej osoby ďalšími osobami. Cieľom *negatívnej identifikácie* je vylúčiť nežiaduci stav, keď jedna osoba využíva identitu viacerých osôb.

2.4 Meranie výkonnosti biometrických metód a zariadení

V biometrických systémoch sa zriedka stretne s dvoma vzorkami jedného užívateľa, ktoré sú úplne rovnaké. Je to dané nie vždy perfektnými podmienkami pri snímaní, zmenami v biometrickej charakteristike osoby, zmenami v okolitom prostredí a zmenami v interakcii osoby so snímačom.

Cieľom biometrického systému je, aby oprávnenej osobe bolo bezchybne umožnené garantované práva. A naopak osoba, ktorá tieto práva nemá, musí byť rovnako bezchybne rozpoznaná a odmietnutá.

V praxi sa používajú dva základne pojmy :

Pravdepodobnosť chybného odmietnutia - False Rejection Rate (FRR)

Pravdepodobnosť chybného prijatia - False Acceptance Rate (FAR)

2.4.1 Pravdepodobnosť chybného odmietnutia (False Rejection Rate – FRR)

Táto veličina udáva, s akou pravdepodobnosťou bude biometrické zariadenie chybovať a odmietne oprávneného užívateľa. V dôsledku toho sa musí užívateľ znova pokúsiť dokázať svoju identitu. Názorná ukážka je na obrázku 2.1.

Pravdepodobnosť chybného odmietnutia FRR je definovaná:

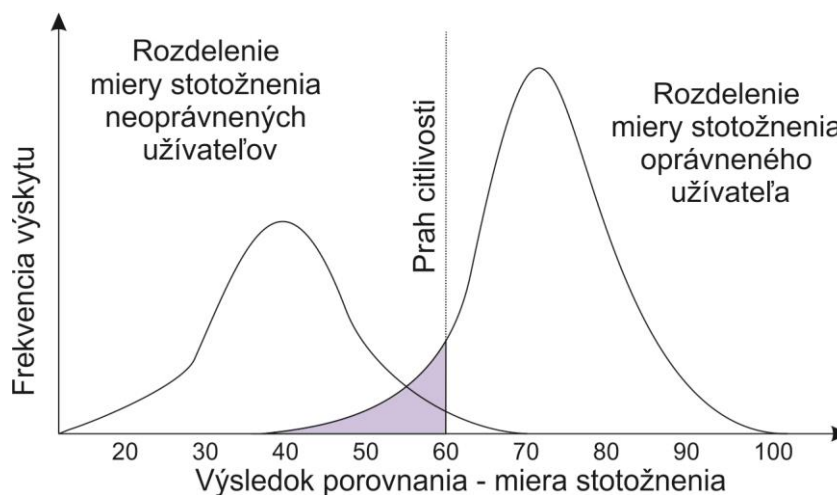
$$FRR = \frac{N_{FR}}{N_{EIA}} \quad \text{alebo} \quad FRR = \frac{N_{FR}}{N_{EVA}},$$

Kde:

N_{FR} – počet chybných odmietnutí (Number of False Rejection)

N_{EIA} – počet pokusov oprávnených osôb o identifikáciu (Number of Enrolle Identification Attempts)

N_{EVA} – počet pokusov oprávnených osôb o verifikáciu (Number of Enrolle Verification Attempts)



Obrázok 2.1 – vymalovaná plocha predstavuje FRR

Z bezpečnostného hľadiska u civilných systémoch sa nejedná o kriticky negatívny jav. Tento jav je nežiaduci len z pohľadu užívateľskej príjemnosti. V praxi nie je žiaduce, aby existoval veľký počet oprávnených osôb, ktoré by boli odmietané. U policajne-súdnych systémoch ide o závažný nedostatok, keďže osoba nemusí byť identifikovaná kvôli chybe systému.

2.4.2 Pravdepodobnosť chybného prijatia (False Acceptance Rate – FAR)

Táto veličina udáva, s akou pravdepodobnosťou bude biometrické zariadenie chybovať a prijme neoprávneného užívateľa. Názorná ukážka je na obrázku 2.2. V dôsledku toho môže dôjsť k nežiaducim aktivitám a následne aj narušeniu majetku, stability objektu, poškodenia práv majiteľa a pod.

Pravdepodobnosť chybného prijatia FAR je definovaná:

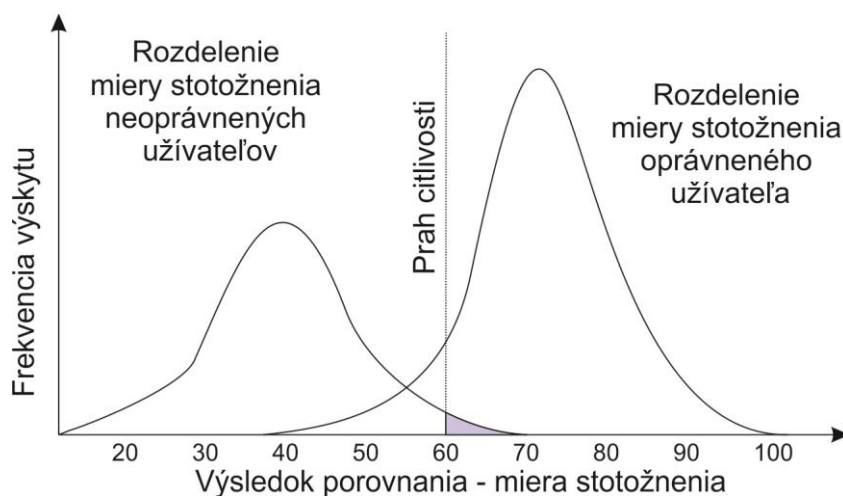
$$FRR = \frac{N_{FA}}{N_{IA}} \quad \text{alebo} \quad FRR = \frac{N_{FA}}{N_{IVA}},$$

Kde:

N_{FA} – počet chybných prijatí (Number of False Acceptance)

N_{IA} – počet pokusov neoprávnených osôb o identifikáciu (Number of Impostor Identification Attempts)

N_{IVA} – počet pokusov neoprávnených osôb o verifikáciu (Number of Impostor Verification Attempts)

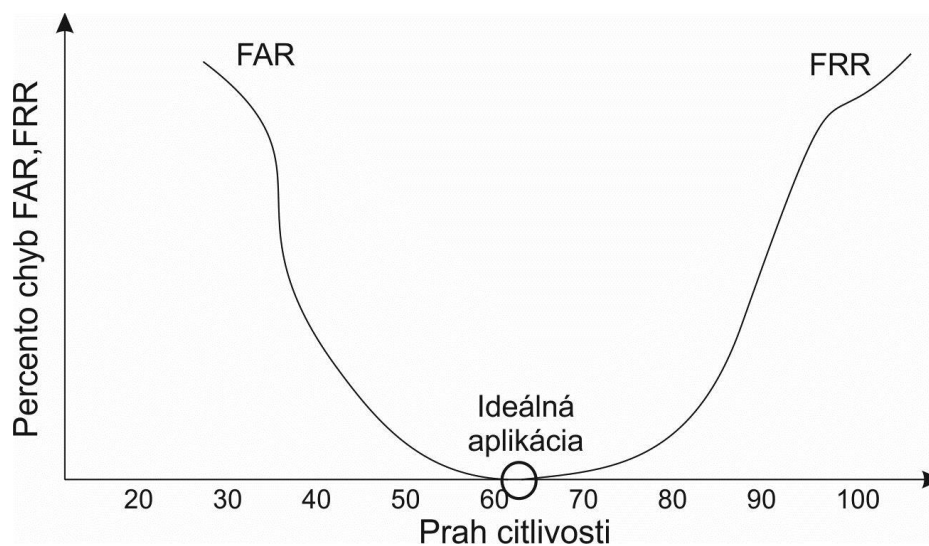


Obrázok 2.2 – vymalovaná plocha predstavuje FAR

2.4.3 Vzťah FRR a FAR

Ideálne zariadenie/systém nevykazuje žiadnu chybovosť, nespoľahlivosť. Všetky osoby sú 100% rozpoznané. V tomto prípade platí teda :

$$FRR = FAR = 0$$



Obrázok 2.3 – ideálna biometrická aplikácia

Znalosť minimálnych a maximálnych hodnôt FRR a FAR je v praxi nepostačujúce. Pre praktické oceňovanie jednotlivých biometrických aplikácií je nutná znalosť celého priebehu oboch kriviek v závislosti na prahu citlivosti. Potom môžeme odčítať dvojice hodnôt FRR a FAR, ktoré majú už

zmysluplnú hodnotu. Pomocou týchto dvojíc sa zostavuje krivka v anglickej terminológii zvaná *Receiver operating characteristic (ROC)*.

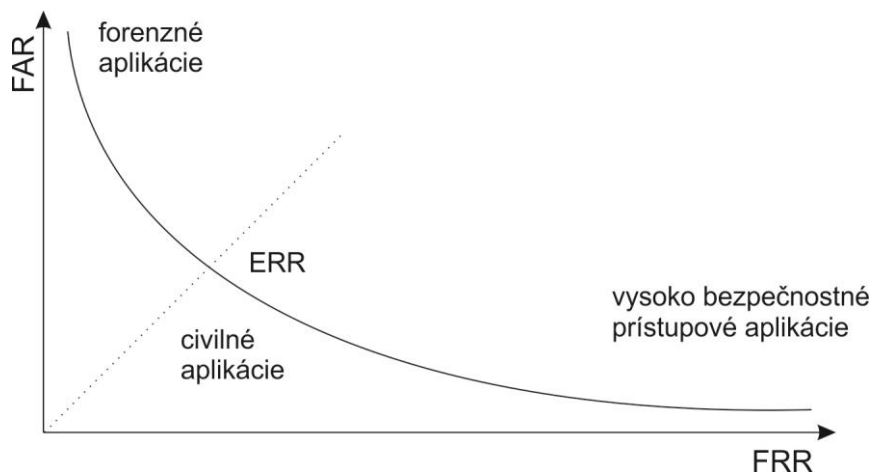
Receiver operating characteristics

Na základe krivky, ktorá určuje vzájomný vzťah pravdepodobností FRR a FAR, môžeme objektívnejšie posudzovať kvalitu biometrických aplikácií. Bod, v ktorom sa krivky pretínajú, sa nazýva *Equal Error Rate (ERR)*. Ukážka ROC krivky je na obrázku 2.4.

Pre tento bod platí rovnosť:

$$FRR_{ERR} = FAR_{ERR}$$

Tento bod slúži iba pre orientačné porovnanie dvoch aplikácií. V praxi záleží na tom, k čomu bude aplikácia použitá. A podľa cieľa použitia je potom nastavený jej prah citlivosti.



Obrázok 2.4 – ROC krivka

2.5 Multi-biometrické systémy

2.5.1 Úvod do multi-biometrických systémov

Multi-biometrické systémy využívajú viacero zdrojov biometrických informácií. To môže byť vykonané buď spojením viacerých biometrických vlastností osoby, alebo viacerými spôsobmi extrakcie informácií a rôznymi porovnávacími algoritmami. Takéto systémy môžu zvýšiť presnosť porovnávania a odraziť viacero útokov.

2.5.2 Taxonómia multi-biometrických systémov

Multi-biometrické systémy závisia na viacerých zdrojoch biometrických informácií osoby. Podľa typu tohto zdroja môžeme rozdeliť multi-biometrické systémy do šiestich kategórií: *multi-sensor*, *multi-algorithm*, *multi-instance*, *multi-sample*, *multimodal* and *hybrid*.

1. Multi-senzor systémy: Tieto systémy využívajú viacero senzorov na získanie jednej biometrickej vlastnosti. Napríklad fotka tváre spolu s infračerveným senzorom, na získanie informácie z pod povrchu tváre.

2. Multi-algorithm systémy: Využívajú rôzne algoritmy pre získanie šablóny z jednej biometrickej vlastnosti a/alebo rôzne porovnávacie algoritmy.

3. Multi-instance systémy: Využívajú viacero príkladov tej istej biometrickej vlastnosti. Napríklad, pravý a ľavý ukazovák osoby.

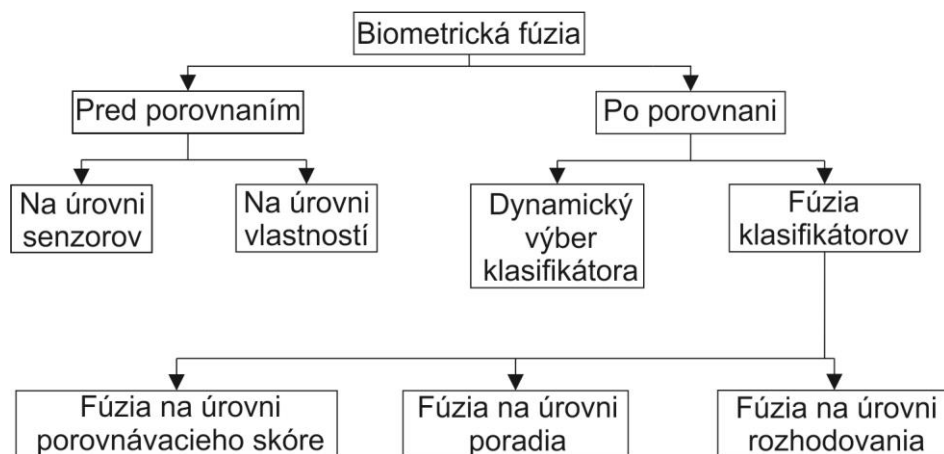
4. Multi-sample systémy: Jeden senzor môže byť využitý na získanie viacerých snímok tej istej biometrickej vlastnosti, za účelom pokrytia menších zmien, ktoré môžu nastať pri danej vlastnosti.

5. Multi-modal systémy: Určujú identitu pomocou viacerých biometrických vlastností. Napríklad tvár a hlas.

6. Hybrid systémy: Využívajú viacero z predchádzajúcich spôsobov.

2.5.3 Úrovnne fúzie

Podľa typu informácie dostupnej v určitom module, môžu byť definované rôzne typy fúzií. Sanderson and Paliwal kategorizujú rôzne úrovne fúzie do dvoch väčších kategórií: pred porovnaním a po porovnaní.



Obrázok 2.5 – Úrovnne fúzie [6]

2.5.3.1 Fúzia na úrovni senzorov

Surové biometrické dáta získané od osoby sú najbohatším zdrojom informácií ale je tiež očakávané, že budú obsahovať šum. Na tejto úrovni sa prevádza usporiadanie surových dát získaných pomocou rôznych senzorov alebo viacerých snímok biometrie získaných jedným senzorom.

2.5.3.2 Fúzia na úrovni vlastností

Na tejto úrovni dochádza k spojeniu jednotlivých biometrických vlastností do jednej pomocou primeraných normalizácií, transformácií a redukcií.

Normalizácia vlastností

Hodnoty vektorov jednotlivých vlastností môžu vykazovať značné rozdiely v ich rozsahu a forme. Jednoduché spojenie nie je vo väčšine prípadov vhodné. Napríklad, keď hodnoty jedného vektora sú v rozsahu $[0,100]$ a hodnoty druhého sú v rozsahu $[0,1]$, potom vzdialenosť medzi dvoma spojenými vektormi bude viac citlivá na prvú vlastnosť ako na druhú. Úlohou normalizácie je modifikovať tieto vektory do spoločnej domény. Existuje viacero možností normalizácie, z tých jednoduchších to sú min-max a mediánova normalizácia.

Výber a transformácia vlastností

Spojením dvoch vektorov vlastností získame nový vektor. Tento vektor však nemusí bezpodmienečne viesť k lepším výsledkom pri porovnaní. Proces výberu vlastností redukuje počet dimenzii vektora. Pri transformácii je vektor namapovaný do nižšej dimenzie.

2.5.3.3 Fúzia na úrovni porovnávacieho skóre

Toto skóre udáva výsledok porovnania dvoch vlastností získaných pomocou rovnakého extraktora rysu. Podobnostné skóre udáva ako „podobné“ sú si jednotlivé vlastnosti, pričom vzdialenostné skóre udáva ako sú „rozdielne“.

Pri tejto fúzii sú jednotlivé skóre získané z rôznych porovnaní kombinované a vytvárajú tak nové skóre, ktoré môže byť následne použité pri rozhodovaní o identite. Metódy fúzie môžeme rozdeliť do troch kategórií: *density-based*, *transformation-based*, *classifier-based*.

Na tejto úrovni sa jednotlivé výsledky porovnaní jednotlivých vlastností spracujú podľa určitých metód a určí sa, či osoba bude prijatá alebo nie. Sú to napríklad tieto metódy: „AND“ a „OR“ pravidlá, „hlasovanie väčšiny“, a iné.

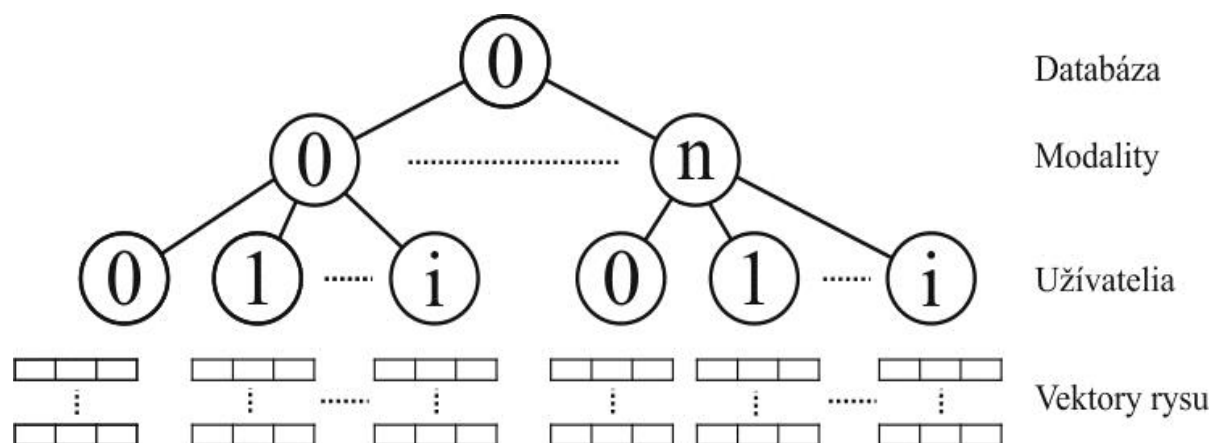
3 Návrh riešenia a implementácia

V tejto kapitole sa pozrieme na návrh a spôsob implementácie jednotlivých častí potrebných k evaluácii multi-biometrického systému. Pozrieme sa na spôsob načítania jednotlivých vektorov rysu a uloženia v aplikácii. Ďalej si popíšeme jednotlivé úrovne fúzie, druhy použitých normalizácií a fúzií. Celá aplikácia je tvorená v jazyku `c++` a využíva knižnicu *Opencv 2.4.9*. Každá časť potrebná pre výpočet je rozdelená do logických celkov, ktoré sú implementované vo forme tried. Tieto triedy obsahujú potrebné funkcie a dátové štruktúry, ktoré sú pre ich funkčnosť potrebné. Tento spôsob som zvolil preto, aby bolo možné jednotlivé časti použiť nezávisle na sebe.

3.1 Načítanie dát

Návrh

Prvou úlohou aplikácie je načítanie dát, ktoré bude spracovávať. Aplikácia je navrhnutá tak, že si jednotlivé vektory rysov uloží do stromovej štruktúry.



Obrázok 3.1 – Stromová štruktúra uloženej databázy

Toto načítanie ale vyžaduje aby bola databáza uložená v podobnej štruktúre priečinkov, to znamená `/databáza/id_modality/id_užívateľa/vektory_rysu`. Toto riešenie je výhodné z hľadiska manuálnej modifikácie databázy, keďže jednotlivé vektory jedného užívateľa pri určitej modalite sú uložené v jednom adresári. Takto organizovaná databáza je výhodná aj pre automatickú modifikáciu databázy, keďže nie je potrebné kontrolovať názvy všetkých súborov. Nevýhoda tohto spôsobu spočíva v tom, že keď je vektor rysu jednej osoby presunutý do priečinka druhej osoby, tak to aplikácia nerozozná.

Implementácia

O načítanie a vytvorenie stromu sa stará trieda `Database`. Štruktúra uzlu vyzerá takto:

```
struct Node
{
    std::wstring someData;           //názov uzlu
    int size;                        //počet pod-uzlov
    std::vector<double> info;        //vektor rysu
    Database::Node **subNodes;      //odkaz na pod-uzol
};
```

Táto trieda obsahuje 3 verejné metódy `load_data`, `makeNode` a `print`. Pomocou metódy `load_data` sa načíta celá databáza, ktorá je metóde predaná ako parameter (adresár, ktorého podadresármi sú adresáre jednotlivých modalít). Táto metóda využíva jednu privátnu metódu triedy, a to `load_file`, ktorá z dát zo zadaného súboru vytvorí vektor.

3.2 Porovnanie a metriky

Návrh

Pre výpočet množín skóre z dát, je potrebné porovnať jednotlivé vektory. Aby sme mohli vypočítať vzájomné skóre dvoch vektorov, potrebujeme metriku. Existuje viacero druhov metrík, medzi tie najzákladnejšie patria *euklidovská vzdialenosť* a *city-block vzdialenosť*:

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \text{ a } d = \sum_{i=1}^k |a_i - b_i|$$

Keď už máme nejakú metriku, tak ju aplikujeme na vektory z istej modality. Pri porovnaní vektorov patriacich jednej osobe získame skóre oprávneného užívateľa (*genuine score*), pri porovnaní vektorov dvoch odlišných osôb dostaneme skóre neoprávneného užívateľa (*impostor score*). Tieto skóre si môžeme uložiť do množín a následne si z nich vypočítať EER.

Implementácia

Metriky a tzv. porovnávač (*matcher*) sú zahrnuté v triede `Metrics`. Zatiaľ sú implementované tri metriky, a to: *euklidovská*, *city-block* a *korelačná* (*correlation distance*). Prvé dve sú implementované na základe vzorcov uvedených vyššie. Korelačná metrika je implementovaná na základe vzorca:

$$d = 1 - \frac{1}{n-1} \sum \left(\frac{x - \bar{x}}{s_x} \right) \left(\frac{y - \bar{y}}{s_y} \right)$$

Kde s_x a s_y sú štandardné odchýlky vektorov x a y . \bar{X} a \bar{Y} predstavujú priemernú hodnotu vektora. Trieda obsahuje porovnávač (*matcher*) a funkciu `lowest_EER_pattern`, ktorá slúži na priradenie metriky ku modalite, tak aby mala čo najnižšie výsledné EER.

Pre porovnanie slúži funkcia `compare`, funkcii sa predá z databázy jedna modalita, a ukazatele na vektory, v ktorých následne budú uložené množiny skóre pre oprávnených a neoprávnených užívateľov (*genuine, impostor scores*) a tiež číslo metriky, ktorá sa ma pre výpočet skóre použiť. Na základe čísla metriky, sa pomocou násobného vetvenia (*switch/case*), sa vytvoria množiny skóre. Je možné do implementovať ďalšie druhy metrík, ale potom je potrebné inkrementovať konštantu `METRICS_COUNT`, a pridať vetvu do násobného vetvenia vo funkcii `compare`.

3.3 EER – Equal Error Rate

Návrh

Ako bolo spomenuté v teoretickej časti, EER je miesto, kde hodnoty FRR a FAR sú rovnaké. Takže na výpočet EER potrebujeme najprv vypočítať hodnoty FRR a FAR. Vzorce na výpočet FRR a FAR sú uvedené v kapitolách 2.4.1 a 2.4.2. Keďže v aplikácii nie sú FRR a FAR spojené, tak ich rovnosť, pri presnom porovnaní, nemusí nikdy nastať. Preto bolo potrebné navrhnuť spôsob, ako zistiť, či sa už približne nerovnajú. O hodnotách FRR a FAR vieme, že keď rastie jedna, druhá klesá. Túto vedomosť môžeme využiť. Preto je aplikácia navrhnutá tak, že si uchováva absolútnu hodnotu rozdielu FRR a FAR a tu porovná s predchádzajúcou. Výpočet pokračuje, kým sa hodnota rozdielu znižuje.

Implementácia

O výpočet a uloženie potrebných informácií o EER sa stará trieda `Eer`. Keďže je potrebné si ponechať informácie o priebehu FRR a FAR obsahuje trieda štruktúru pre tieto informácie:

```
struct eer_return
{
    double eer_value;           //hodnota EER
    double threshold;          //hodnota prahu pre EER
    std::vector<double> fmr;     //vektor priebehu FMR
    std::vector<double> fnmr;    //vektor priebehu FNMR
    std::vector<double> thresholds; //vektor prahov pre FMR a FNMR
};
```

Pre výpočet FRR a FAR sú v triede dve funkcie `False_Acceptance_Rate` a `False_Rejection_Rate`. Týmto funkciám je predaný vektor so skóre neoprávnených resp. oprávnených užívateľov a prah, pre ktorý sa má hodnota vypočítať. Pre naplnenie hore uvedenej štruktúry slúži funkcia `EER`. Tejto funkcii sú predané množiny skóre oprávnených a neoprávnených užívateľov. Tieto množiny sa spoja do nového vektora a jeho hodnoty slúžia ako prahy pre jednotlivé výpočty FAR a FRR. Výpočet EER prebieha takto:

```

e_pom = abs((false_acc_rate - false_rec_rate)); //rozdiel FAR a FRR
if (e_pom <= epsilon) //rozdiel menší ako redchádzajúci
{
    epsilon = e_pom; //nová hodnota rozdielu
    min_eer.threshold = scores[i]; //nová hodnota prahu
    min_e_rate = false_acc_rate; //nová hodnota EER
}

```

Pri každej iterácii sa naplňajú informácie v hore uvedenej štruktúre, z ktorých sa dajú získať informácie o priebehu FAR a FRR, z ktorých sa dá následne vykresliť graf. Pred uložením týchto hodnôt, sú vektory ešte usporiadané a to tak, že FNMR zostupne a FMR a thresholds vzostupne.

3.4 Práca s vektormi

Návrh

Kvôli normalizácii dát, je potrebné získať nejaké štatistické koeficienty z týchto dát, ako sú napríklad minimum, maximum, priemerná hodnota, medián a smerodajná odchýlka. Keďže väčšina dát v aplikácii je uložená vo forme vektorov, bola potrebné navrhnuť triedu, ktorá je schopná tieto dáta získať.

Implementácia

V aplikácii sa o výpočet týchto štatistických koeficientov stará trieda VEKTOR. Obsahuje funkcie `v_min`, `v_max`, `median`, `mad`, `mean` a `stdDeviation`. Ktoré zo zadaného vektora vypočítajú Minimum, maximum, medián, mediánovu absolútnu odchýlku, priemernú hodnotu a smerodajnú odchýlku. Ďalšia funkcia v tejto triede sa nazýva `lowest_colleration`. Táto funkcia zistí, ktoré dva vektory z množiny vektorov majú voči sebe najmenšiu koreláciu. Táto funkcia má využitie pri zisťovaní potencionálnych kombinácii modalít pre lepší výsledok.

3.5 Normalizácia

Návrh

Pre použitie fúzie na úrovni vlastností a skóre (*feature-level*, *score-level*), je potrebná normalizácia aby sa mohli jednotlivé hodnoty spolu kombinovať. Existuje množstvo normalizačných metód pričom každá má iné vlastnosti (tab. č. 3.1). Medzi základné patrí min-max normalizácia, ktorá má vzorec:

$$s' = \frac{s - \min\{s_k\}}{\max\{s_k\} - \min\{s_k\}}$$

Ďalšou jednoduchou metódou je decimálne škálovanie, ktorá má vzorec:

$$s' = \frac{s}{10^n}$$

$$n = \log_{10} \max\{s_k\}$$

Pričom $\{s_k\}$ je množina všetkých hodnôt.

| Normalizačná technika | Robustnosť | Efektívnosť |
|-----------------------|------------|-------------|
| Min-max | Nie | Vysoká |
| Decimálne škálovanie | Nie | Vysoká |
| Z-score | Nie | Vysoká |
| Medián a MAD | Áno | Stredná |
| Dvojitá sigmoidálna | Áno | Vysoká |
| Tanh | Áno | Vysoká |

Tabuľka 3.1- Zobrazuje vlastnosti jednotlivých normalizácií [5]

Implementácia

Normalizačné metódy sú implementované v triede **Normalization**. Trieda obsahuje štruktúru, do ktorej sa ukladajú normalizačné koeficienty z trénovacích dát.

```
struct normalization_data{
    vector<double> feature_mean;
    vector<double> feature_std;
    vector<vector<vector<double>>> vector_of_genuine_scores;
    vector<vector<vector<double>>> vector_of_impostor_scores;
};
```

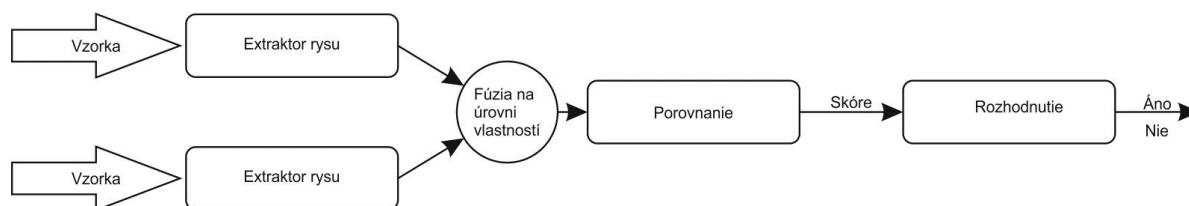
V tejto štruktúre sa nachádzajú hodnoty, ktoré sú vypočítané pomocou funkcie **normalization_coef**. Táto funkcia pre každú modalitu zo zadanej databázy vypočíta všetky množiny skóre pomocou všetkých implementovaných metrík. Z týchto množín následne vypočíta priemernú hodnotu a hodnotu smerodajnej odchýlky.

Jednotlivé normalizačné metódy sú implementované vo funkciách **min-max**, **median** a **z-score**. Funkcie **min-max** a **median** znormalizujú zadaný vektor. Funkcia **z-score** znormalizuje len zadanú hodnotu pomocou zadaných normalizačných koeficientov.

3.6 Fúzia na úrovni vlastností (*feature-level*)

Návrh

Pri tejto úrovni fúzie sa jednotlivé modality kombinujú ešte pred porovnaním (obrázok 3.1). To znamená, že sa spájajú jednotlivé vektory rysov daného užívateľa. Táto úroveň fúzie je nevrhnutá tak, že buď sa jednotlivé vektory jednoducho konkatenujú alebo sa využíva metóda PCA [4]. Potom sa jednotlivé výsledné vektory porovnávajú pomocou nejakej metriky, z čoho získame množinu výsledných skóre. Z tejto množiny potom môžeme vypočítať EER a priebeh FAR a FRR.



Obrázok 3.1 – schéma fúzie na úrovni vlastností

Implementácia

Pre fúziu na úrovni vlastností slúži trieda `Feature_level`. Táto trieda obsahuje privátnu funkciu `concatenation`, a verejné funkcie `train` a `fusion`.

Pre uloženie dát z trénovacej funkcie slúži štruktúra:

```

struct feature_train_data{
    PCA pca;
    vector<double> min;
    vector<double> max;
    vector<double> mean;
    vector<double> std;
    int normalization_metod = -1;
    int fusion_metod = -1;
    int metric = -1 ;
};
  
```

Funkcia `concatenation` vytvorí „novú modalitu“, ktorej vektory rysu budú pozostávať z konkatenancie pôvodných. To znamená, že ak máme databázu, ktorá obsahuje, napríklad 3 modalitty, tak po prevedení tejto funkcie získame databázu s jednou modalitou. Funkcia má parameter, ktorým môžeme určiť, ktoré modalitty chceme skombinovať. Ak tento parameter nie je zadaný funkcia skombinuje všetky modalitty.

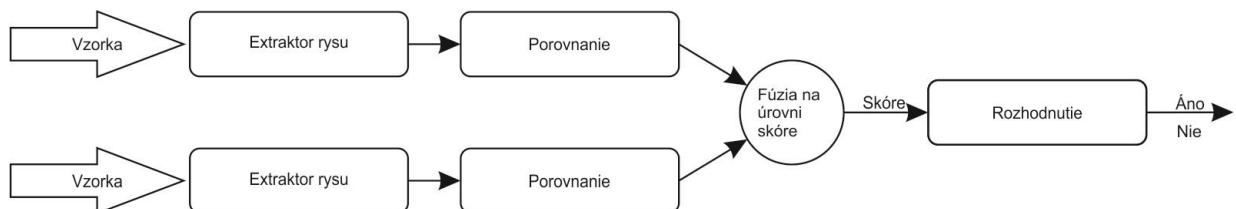
Funkcia `train` slúži na natrénovanie fúzie a získanie normalizačných koeficientov, keď sa jedná o fúziu za pomoci PCA. V režime trénovania funkcia skonkatenuje všetky modalitty z trénovacích dát, čím získame nové vektory rysov. Parameter `feature_fusion_method` určuje, ktorá metóda fúzie sa použije. Ak je nastavená základná hodnota tohto parametra, vykoná sa trénovanie všetkých implementovaných metód fúzie. Ak bola nastavená iná hodnota, tak sa vykoná tréning odpovedajúcej metódy. Parameter `metric` udáva, ktorá metrika sa má použiť. Ak je nastavená základná hodnota EER trénovacích at sa vypočíta pre všetky metriky a vyberie sa metrika, s ktorou bolo dosiahnuté najnižšie EER. Pri použití PCA sa musia dáta hodnoty vektora znormalizovať. Využívajú sa na to implementované normalizačné metódy a to z-score a min-max, z ktorých sa potom automaticky vyberie tá z ktorou boli dosiahnuté lepšie výsledky. Hodnoty pre PCA projekciu sa uložia do súboru „PCA_data.txt“ a hodnoty pre normalizáciu do súboru „Normalizacne_data.txt“. Výsledkom tejto funkcie je spôsob akým má byť fúzia prevedená aby sme teoreticky dosiahli najnižšie EER na testovacích dátach.

Funkcia `fusion` slúži na prevedenie fúzie, ktorá bola natrénovaná alebo presne zadaná užívateľom. Výsledný spôsob prevedenia fúzie je vypísaný na štandardný výstup spolu s výsledným EER. Priebiehy FNMR a FMR spolu s odpovedajúcimi prahmi sú uložené v súbore „FNMR_FMR_P.txt“.

3.7 Fúzia na úrovni skóre (*score-level*)

Návrh

Na tejto úrovni sa kombinujú jednotlivé množiny skóre, ktoré sme získali porovnaním v rámci jednotlivých modalít (obrázok 3.2). Predtým ako ich ale môžeme skombinovať, musíme jednotlivé množiny skóre znormalizovať. Pre kombináciu znormalizovaných skóre sa využíva viacero pravidiel. Medzi tie základné patria *sum-rule*, *min-rule* a *max-rule*. Z výslednej množiny skóre sa následne vypočíta EER a priebeh FMR a FNMR.



Obrázok 3.2 – schéma fúzie na úrovni skóre

Implementácia

Pre fúziu na úrovni skóre slúži trieda `Score_level`. Trieda obsahuje privátnu funkciu `score_fusion` a dve verejné `train` a `fusion`.

Funkcia `train` slúži na natrénovanie spôsobu fúzie. Najprv sa znormalizujú jednotlivé množiny skóre. Tieto normalizačné koeficienty sa uložia do súboru „Normalizacne_data.txt“. Následne sa pre všetky implementované pravidlá fúzie volá funkcia `score_fusion`, ktorá vráti kombináciu modalít a EER, ktoré touto kombináciou pre dané pravidlo dosiahneme. Takto po využití všetkých implementovaných pravidiel (*sum-rule*, *max-rule*, *min-rule*), získame informácie o tom, ktoré pravidlo a modalita využít, aby sme dosiahli pravdepodobne najnižšie EER.

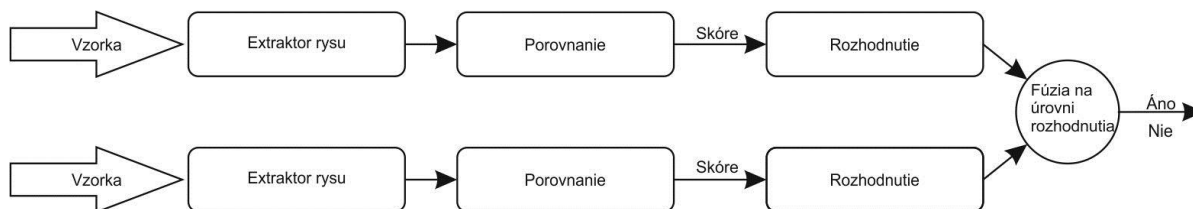
Funkcia `score_fusion`, vyberá kombináciu modalít nasledujúcim spôsobom. Najprv zo všetkých vektorov skóre vyberie dva, ktoré majú voči sebe najmenšiu koreláciu. Potom pre túto dvojicu skombinuje pomocou zadaného pravidla, vypočíta EER a následne porovná so zvyšnými vektormi a vyberie ten, ktorý má s ním najnižšiu koreláciu. Nasleduje skombinovanie vektorov, a výpočet EER a porovnanie s predchádzajúcim EER. Toto sa opakuje kým EER klesá alebo máme ešte nejaké vektory skóre, ktoré by sme mohli skombinovať.

Funkcia `fusion` slúži na prevedenie fúzie, ktorá bola natrénovaná alebo presne zadaná užívateľom. Výsledný spôsob prevedenia fúzie je vypísaný na štandardný výstup spolu s výsledným EER. Priebehy FNMR a FMR spolu s odpovedajúcimi prahmi sú uložené v súbore „FNMR_FMR_P.txt“.

3.8 Fúzia na úrovni rozhodovania (*decision-level*)

Návrh

Na tejto úrovni sa kombinujú rozhodnutia u jednotlivých modalít (obrázok 3.3). Medzi základne pravidla kombinovania rozhodnutí patria pravidla AND a OR. Pri pravidle AND musia všetky rozhodnutia byť v prospech užívateľa aby bol prijatý, pri pravidle OR naopak stačí len jedno rozhodnutie v prospech užívateľa aby bol prijatý.



Obrázok 3.3 – schéma fúzie na úrovni rozhodovania

Implementácia

Pre fúziu na tejto úrovni slúži trieda `Decision_level`. Trieda obsahuje privátnu funkciu `decision_fusion` a dve verejné funkcie `train` a `fusion`.

Funkcia `train` slúži na natrénovanie fúzie na tréovacích dátach. Po získaní vektorov skóre pre jednotlivé modalitty, sú tieto jednotlivé skóre transformované na 0 a 1, kde 0 znamená prijatý, 1 neprijatý. Toto rozhodnutie sa vykonáva na základe prahu pre EER danej modalitty. Po transformácii jednotlivých vektorov, sa vyberú dva, ktoré majú voči sebe najnižšiu koreláciu. Tieto dva sa spoja do jedného pomocou implementovaného pravidla. Následne sa nájde ďalší vektor skóre, ktorý má najnižšiu koreláciu voči predchádzajúcemu a znova sa spoja do jedného pomocou rovnakého pravidla. Po každom spojení sa vypočíta EER a porovná s predchádzajúcim. Kombinovanie končí vtedy, keď EER prestane klesať alebo nebude už žiadny ďalší vektor na výber. Proces kombinovania vektorov sa opakuje pre všetky implementované pravidlá pre fúziu na úrovni rozhodovania. Výsledkom tejto funkcie, je kombinácia modalít a pravidlo fúzie, pre ktoré bolo EER na tréovacích dátach najnižšie.

Funkcia `fusion` slúži na prevedenie fúzie, ktorá bola natrénovaná alebo presne zadaná užívateľom. Výsledný spôsob prevedenia fúzie je vypísaný na štandardný výstup spolu s výsledným EER. Priebehy FNMR a FMR spolu s odpovedajúcimi prahmi sú uložené v súbore „FNMR_FMR_P.txt“.

3.9 Výsledná aplikácia

Návrh

Aplikácia zo vstupných trénovacích dát vytvorí ideálny postup pre fúziu a následne tento postup aplikuje na testovacích dátach. Výstupom sú informácie o spôsobe fúzie, dosiahnuté EER touto fúziou na testovacích dátach a priebeh FAR a FRR.

Implementácia

Aplikácia sa ovláda pomocou príkazového riadku, teda pomocou parametrov, ktoré su aplikácii predané. Minimálne musí byť zadaný typ fúzie, trénovacie a testovacie dáta. Zvyšné parametre sa natrénuju. Ak užívateľ zadá nejaký parameter, ktorý by sa inak musel natréňovať, tak k natréňovaniu nedôjde a použije sa užívateľom zadaná hodnota.

4 Experimenty

V tejto kapitole budú predstavené dátové súbory použité v experimentoch, bude popísaný priebeh a sledované vlastnosti.

4.1 Biometrická databáza

Na testovanie a experimentovanie s parametrami aplikácie bola použitá jedna databáza, ktorej veľkosť bola pri jednotlivých experimentoch menená. Táto databáza bola poskytnutá vedúcim bakalárskej práce. Pôvodná databáza bola rozdelená na tréningovú časť, kde bolo použitých 197 snímok a testovaciu časť, kde bolo použitých zvyšných 1465 snímok u každej z 21 modalít. Kvôli dĺžke výpočtu pri takto veľkom počte snímok, bol ich počet pre experimentálne účely zredukovaný. Veľkosť tréningovej časti bola zredukovaná na 65 snímok, a počet modalít bol zredukovaný na prvých 6 a počet snímok v testovacej databáze na 197. Počet snímok a veľkosť vektoru je pre každú modalitu uvedená v tabuľke (tab. č.4.1).

| Modalita | Počet snímok- tréningové | Počet snímok- testovacia | Dĺžka vektora snímky |
|----------|--------------------------|--------------------------|----------------------|
| 0 | 65 | 197 | 268 |
| 1 | 65 | 197 | 270 |
| 2 | 65 | 197 | 300 |
| 3 | 65 | 197 | 300 |
| 4 | 65 | 197 | 322 |
| 5 | 65 | 197 | 322 |

Tabuľka 4.1 – Počet snímok a ich veľkosť jednotlivých modalít v testovanej databáze

4.2 Použitý HW

Všetky experimenty boli prevádzané na PC s procesorom Intel Core i7-2630QM, 8GB operačnej pamäte a operačným systémom Windows 7 64b.

4.3 Fúzia na úrovni vlastností

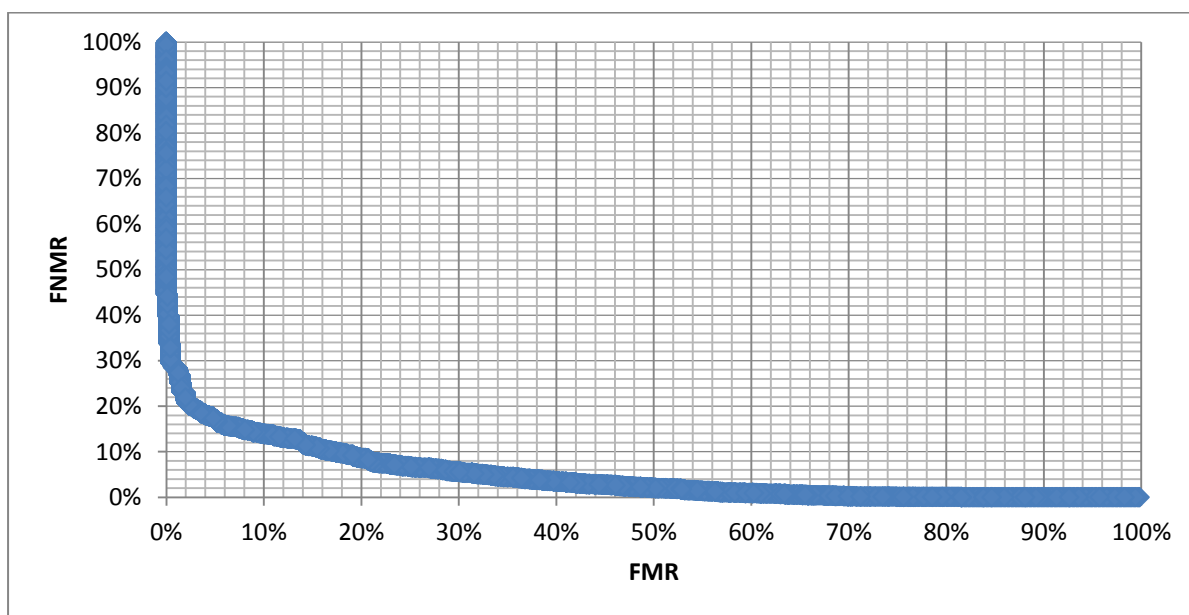
Automatický výber fúzie

Ako prvé bola aplikácia spustená bez parametrov, takže sa vypočítal optimálny výber modalít do fúzie zo zadaných tréningových dát. Tieto informácie boli následne použité pre evaluáciu databázy

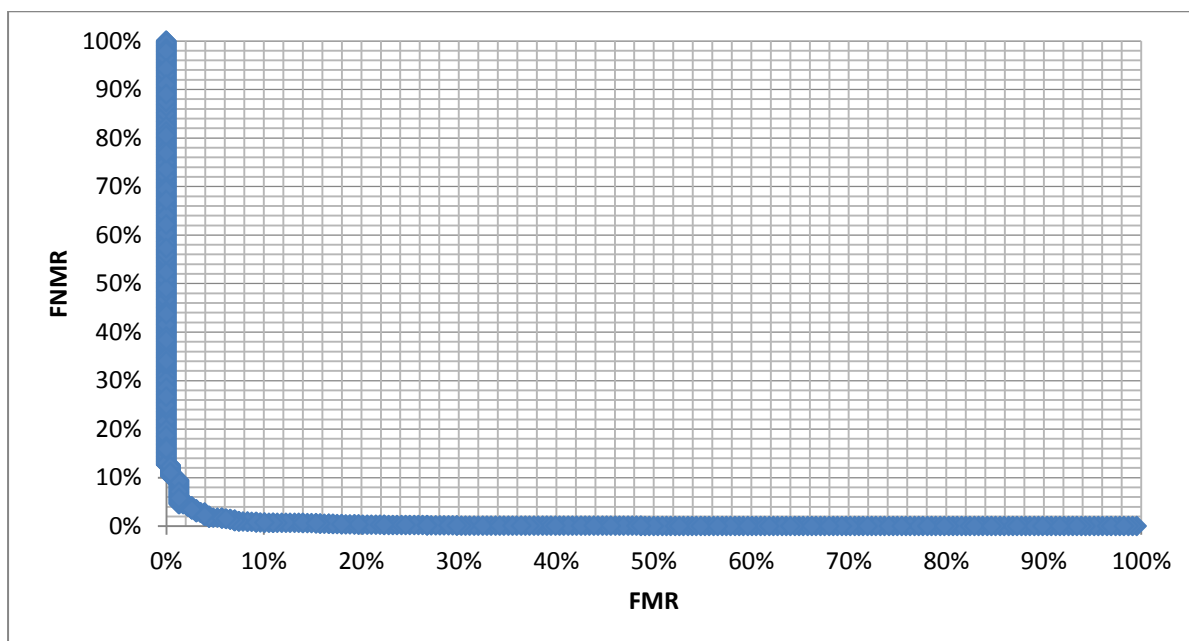
zadanej pre testovanie. Použitím tohto spôsobu kombinácie modalít a využitých techník by sme mali získať najlepšie výsledné EER testovanej databázy.

Aplikácia navrhla skombinovať všetky modalít. Jednotlivé vektory rysu zkonkaténovať a znormalizovať pomocou *z-score* normalizácie. Následne na znormalizované skóre je aplikovaná metóda *PCA*. Výsledné vektory rysov sa porovnávajú pomocou *korelačnej* metriky. Výpočet trval necelé 2 minúty.

Touto kombináciou bolo na tréningových dátach dosiahnuté $EER = 0.0306893$ a na testovacích dátach $EER = 0.128595$. Na grafe 4.1 sa nachádza ROC krivka testovanej databázy. Na grafe 4.2 sa nachádza ROC krivka tréningových dát.



Graf 4.1 – ROC krivka testovacích dát



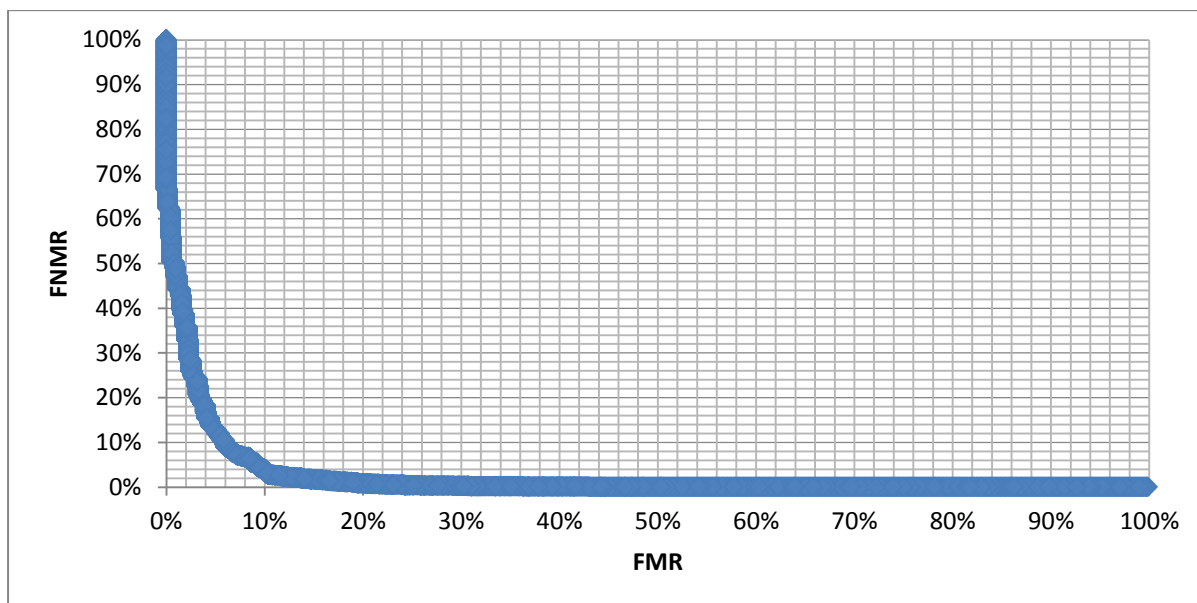
Graf 4.2 – ROC krivka tréningových dát

Automatický výber metódy fúzie

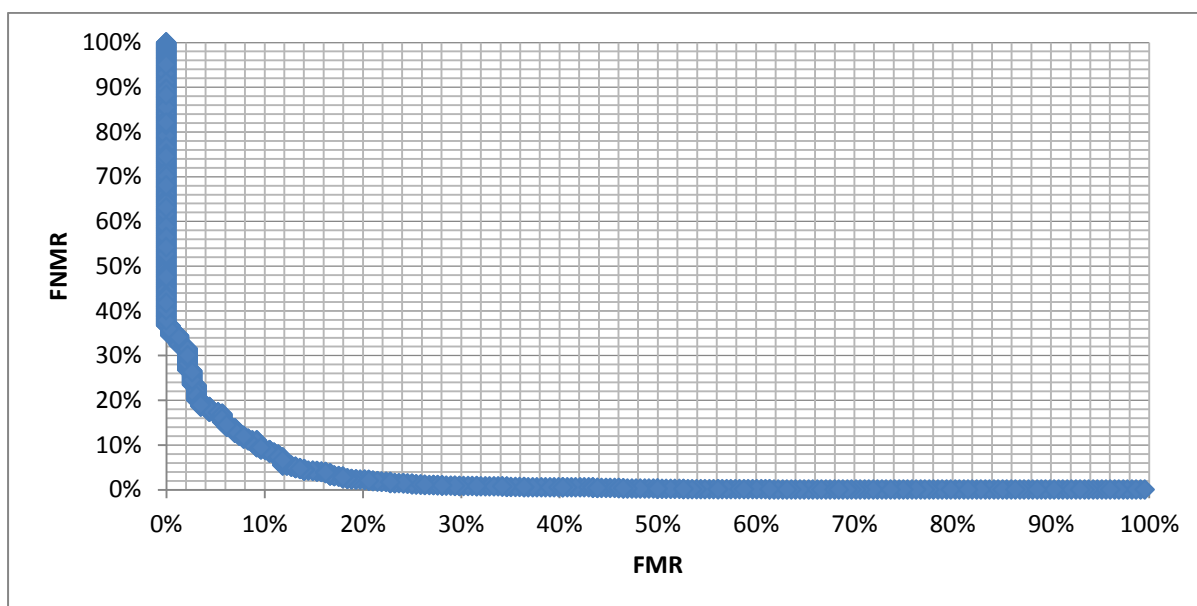
Pri tomto experimente bol na vstupe zadaný typ metriky, ktorý sa použije pri fúzii. Bola vybratá metrika *city-block*. Aplikácia už nebude hľadať ideálnu metriku ale použije zadanú metriku a dopočíta len informácie, ktoré chýbajú. Takže navrhne kombináciu jednotlivých modalít a metódu fúzie, pri ktorej by sme mali získať najlepší výsledok.

Aplikácia navrhla zkonkaténovať všetky modalít a takto vytvorené vektory porovnať pomocou zadanej *city-block* metriky. Výpočet trval minútu a 8 sekúnd.

Touto kombináciou bolo na tréningových dátach dosiahnuté $EER = 0.094428$ a na testovacích dátach $EER = 0.0720391$. Na grafe 4.3 sa nachádza ROC krivka testovanej databázy a na grafe 4.4 ROC krivka tréningovej databázy.



Graf 4.3 – ROC krivka testovacích dát



Graf 4.4 – ROC krivka tréningových dát

Automatický výber metriky

Pri tomto experimente bola na vstupe zadaná metóda prevedenia fúzie. V tomto prípade som zvolil metódu PCA, keďže v predchádzajúcom experimente bola vybratá konkatenancia. Aplikácia sama zvolí druh metriky a spôsob normalizácie dát (*z-score, min-max*).

Aplikácia navrhla použiť *z-score* normalizáciu a *korelačnú* metriku. Výsledok je rovnaký ako pri automatickom výbere všetkých parametrov. Výpočet ale trval iba necelú minútu.

Touto kombináciou bolo na testovacích dátach $EER = 0.128595$ a na trénovacích dátach dosiahnuté $EER = 0.0306893$.

Vplyv zväčšenia databáze na výsledok

Zvýšenie počtu modalít pri zachovaní počtu vektorov pri jednotlivých modalitách

Automatický výber fúzie - Došlo k zmene kombinácie. Aplikácia navrhla využiť všetky modalities, dáta znormalizovať pomocou min-max, použiť PCA metódu a korelačnú metriku. Na testovaných dátach bola dosiahnutá hodnota $EER = 0.0855868$ čo je zlepšenie oproti predchádzajúcej hodnote. Na trénovacích dátach bola dosiahnutá hodnota $EER = 0.0174693$, čo je tiež lepšia hodnota. Rýchlosť výpočtu sa v podstate nezmenila.

Automatický výber metódy fúzie – Nedošlo k zmene kombinácie. Na testovaných dátach bola dosiahnutá hodnota $EER = 0.0709102$ čo je miernym zlepšením oproti predchádzajúcej hodnote. Na trénovacích dátach bola dosiahnutá hodnota $EER = 0.0892351$, čo je tiež lepšia hodnota. Rýchlosť výpočtu sa spomalila o približne 20 sekúnd.

Automatický výber metriky - Došlo k zmene kombinácie. Aplikácia navrhla využiť všetky modalities, dáta znormalizovať pomocou min-max, použiť PCA metódu a korelačnú metriku. Na testovaných dátach bola dosiahnutá hodnota $EER = 0.0855868$ na trénovacích dátach bola dosiahnutá hodnota $EER = 0.0174693$. Rýchlosť výpočtu ostala nezmenená.

Zvýšenie počtu vektorov pre jednotlivé modalities

Počet vektorov pre trénovanie zvýšený na 160 a počet vektorov pre testovanie zvýšený na 380.

Automatický výber fúzie - Došlo k zmene kombinácie. Aplikácia navrhla využiť všetky modalities, dáta znormalizovať pomocou min-max, použiť PCA metódu a korelačnú metriku. Na testovaných dátach bola dosiahnutá hodnota $EER = 0.121303$ čo je mierne zhoršenie oproti pôvodnej hodnote. Na trénovacích dátach bola dosiahnutá hodnota $EER = 0.0340299$, čo je tiež mierne zhoršenie. Dĺžka výpočtu sa zvýšila na 17 minút.

Automatický výber metódy fúzie – Nedošlo k zmene kombinácie. Na testovaných dátach bola dosiahnutá hodnota $EER = 0.0703122$ čo je miernym zlepšením oproti predchádzajúcej hodnote. Na trénovacích dátach bola dosiahnutá hodnota $EER = 0.105476$. Výpočet trval približne 16 minút.

Automatický výber metriky - Došlo k zmene kombinácie. Aplikácia navrhla využiť všetky modalitty, dáta znormalizovať pomocou min-max, použiť PCA metódu a korelačnú metriku. Na testovaných dátach bola dosiahnutá hodnota EER = 0.121303, čo je mierne zlepšenie a na tréningových dátach bola dosiahnutá hodnota EER = 0.0340299. Dĺžka výpočtu sa zvýšila na 17 minút.

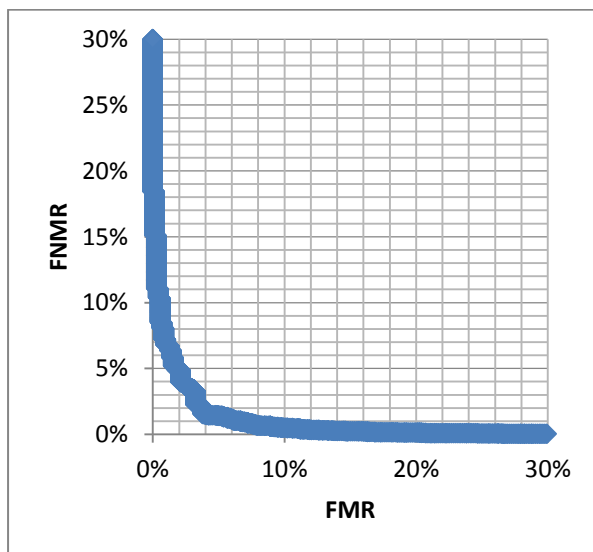
4.4 Fúzia na úrovni skóre

Automatický výber fúzie

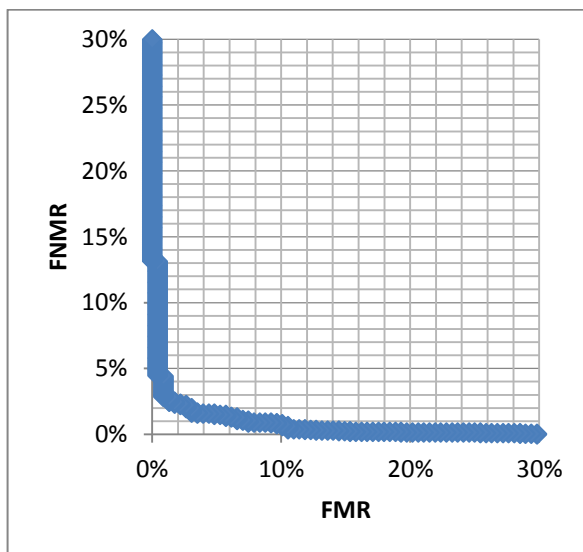
Aplikácia navrhne ideálnu kombináciu pre fúziu skóre. Použitím tohto spôsobu kombinácie modalít a využitých techník by sme mali získať najlepšie výsledné EER testovanej databázy.

Aplikácia navrhla skombinovať modalitty 0,1,2,3,5. Pre jednotlivé modalitty navrhla metriky s najnižším výsledným EER, a to *city-block* metriku pre modalitu číslo 3 a *korelačnú* metriku pre ostatné modalitty. Pre fúziu bolo navrhnuté *sum-rule* pravidlo. Výpočet trval necelé 2 minúty a 20 sekúnd.

Touto kombináciou bolo na tréningových dátach dosiahnuté EER = 0.0221907 a na testovacích dátach EER = 0.0326327. Na grafe 4.5 sa nachádza ROC krivka testovanej databázy. Na grafe 4.6 sa nachádza ROC krivka tréningových dát.



Graf 4.3 – ROC krivka testovacích dát



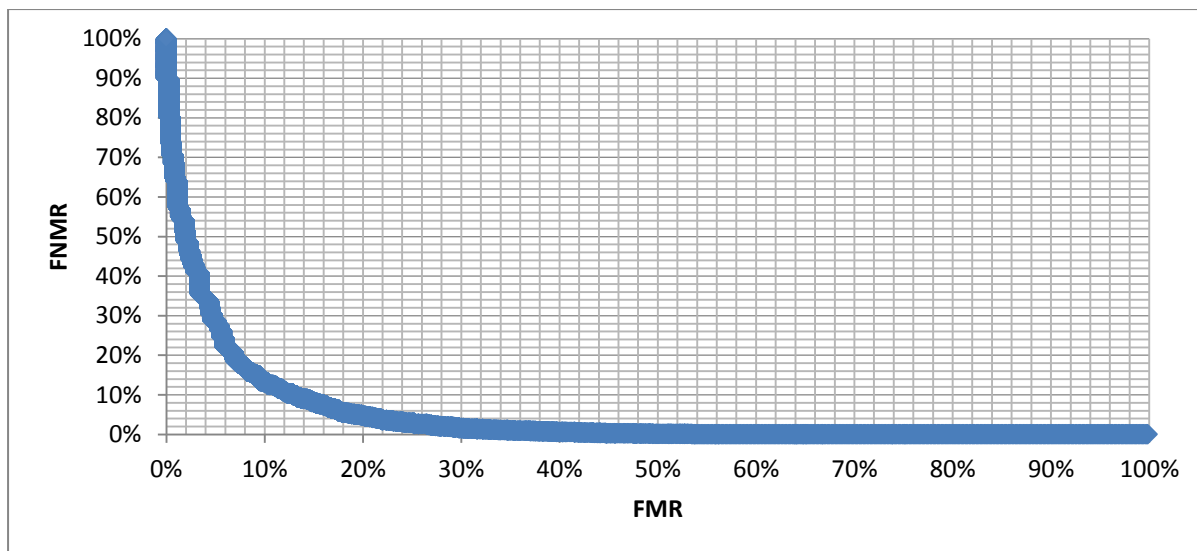
Graf 4.6 – ROC krivka tréningových dát

Automatický výber metódy fúzie

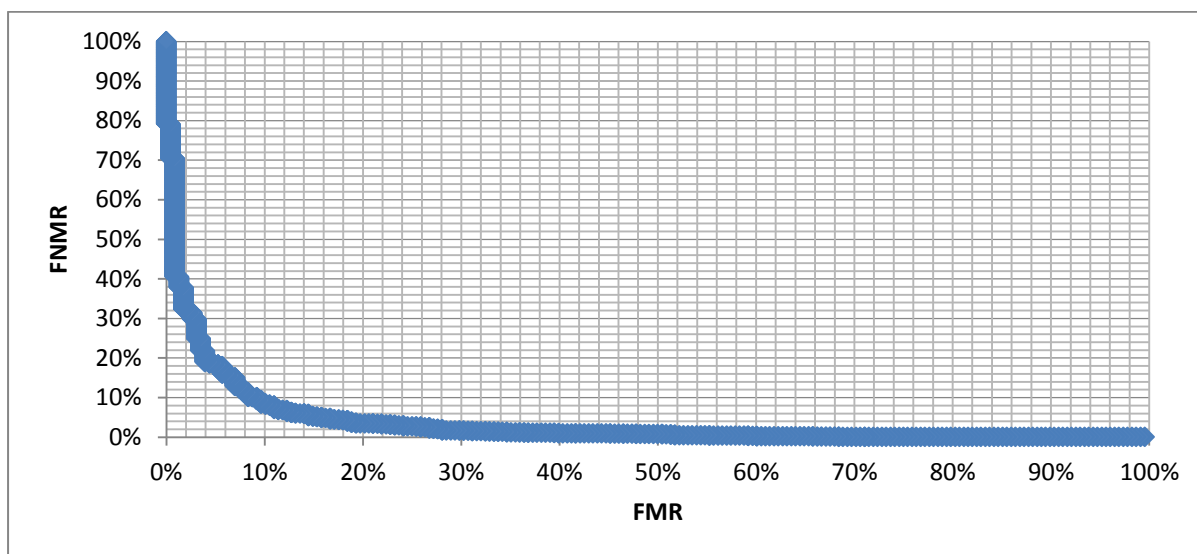
Pri tomto experimente boli zadané metriky pre jednotlivé modalitty. Pre modalitty 0,1 metrika *city-block*, pre modalitty 2,3 *korelačná* metrika, pre modalitty 4,5 *euclidovská* metrika. Aplikácia už nebude hľadať ideálne metriky ale použije zadané metriky a dopočíta len informácie, ktoré chýbajú. Takže navrhne kombináciu jednotlivých modalít a pravidlo fúzie, pri ktorej by sme mali získať najlepší výsledok.

Aplikácia navrhla skombinovať modality 0,1,2,4,5 použitím pravidla *sum-rule*. Výpočet trval minútu a 30 sekúnd.

Touto kombináciou bolo na tréningových dátach dosiahnuté $EER = 0.0934844$ a na testovacích dátach $EER = 0.115639$. Na grafe 4.7 sa nachádza ROC krivka testovanej databázy a na grafe 4.8 ROC krivka tréningovej databázy.



Graf 4.7 – ROC krivka testovacích dát



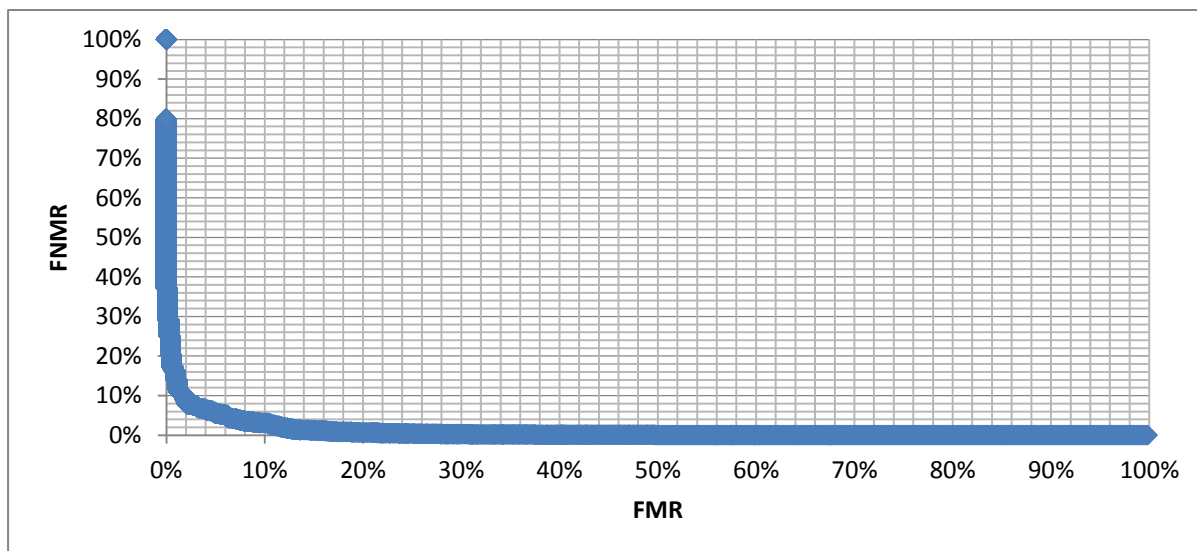
Graf 4.8 – ROC krivka tréningových dát

Automatický výber metriky

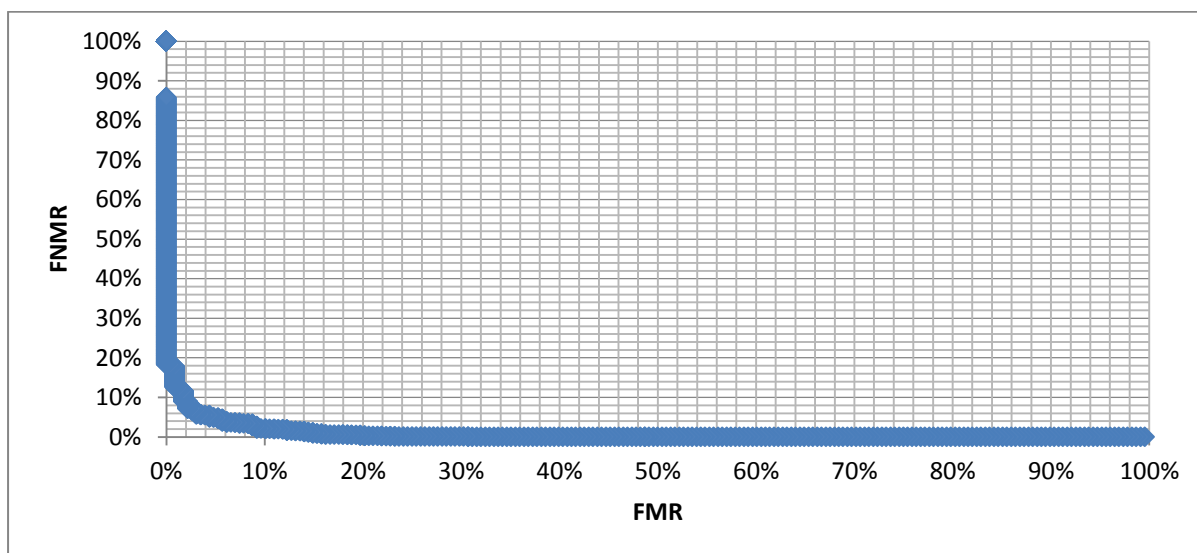
Pri tomto experimente bolo na vstupe zadané pravidlo pre fúziu. Zvolil som pravidlo *min-rule*.

Aplikácia navrhla použiť *city-block* metriku pre modality 2,3, pre ostatné bola zvolená *korelačná* metrika. Kombinovali sa modality 0,1,2,3,5. Výpočet trval približne 2 minúty.

Touto kombináciou bolo na testovacích dátach $EER = 0.0539218$ a na tréningových dátach dosiahnuté $EER = 0.0486308$. Na grafoch 4.9 a 4.10 sú zobrazené jednotlivé ROC krivky.



Graf 4.9 – ROC krivka testovacích dát



Graf 4.10 – ROC krivka trénovacích dát

Vplyv zväčšenia databáze na výsledok

Zvýšenie počtu modalít pri zachovaní počtu vektorov pri jednotlivých modalitách

Automatický výber fúzie – Na modalitách 0,1,2,3,6,7,9,10,11 bola použitá *korelačná* metrika a na modalitách 4,5,8 bola použitá *city-block* metrika. Skóre bolo kombinované z modalít 1,2,5,7,9,10. Bolo použité pravidlo *sum-rule*. Na testovaných dátach bola dosiahnutá hodnota EER = 0.036826 čo je mierne zhoršenie oproti pôvodnej hodnote. Na trénovacích dátach bola dosiahnutá hodnota EER = 0.0193579, čo je lepšia hodnota. Výpočet trval 4 a pol minúty, čo je dvojnásobne dlhšia doba.

Automatický výber metódy fúzie – Pri tomto experimente boli zadané metriky pre jednotlivé modality. Pre modality 0,1,2,3 metrika *city-block*, pre modality 4,5,6,7 *korelačná* metrika, pre modality 8,9,10,11 *euklidovská* metrika. Kombinovali sa modality 1,4,5,6,7,9. Použité pravidlo *sum-rule*. Na testovaných dátach bola dosiahnutá hodnota EER = 0.0780066 čo je miernym zlepšením oproti

predchádzajúcej hodnote. Na tréningových dátach bola dosiahnutá hodnota $EER = 0.0439093$, čo je tiež lepšia hodnota. Výpočet trval 2 minúty a 6 sekúnd.

Automatický výber metriky – Pre modalít 0,1,2,3,6,7,9,10,11 bola použitá *korelačná* metrika, pre zvyšné *city-block* metrika. Kombinovali sa modalít 0,1,2,5 pomocou zadaného *min-rule* pravidla. Na testovaných dátach bola dosiahnutá hodnota $EER = 0.0950487$ čo je zhoršenie oproti pôvodnej hodnote. Na tréningových dátach bola dosiahnutá hodnota $EER = 0.0377715$, čo je lepšie ako pôvodná hodnota. Výpočet trval 3 minúty 20 sekúnd.

Zvýšenie počtu vektorov pre jednotlivé modalít

Počet vektorov pre tréningovanie zvýšený na 160 a počet vektorov pre testovanie zvýšený na 380.

Automatický výber fúzie - Došlo k zmene odporúčaných metrík pre jednotlivé modalít. Modalít 0,1,4,5 používajú *korelačnú* metriku, zvyšné *city-block* metriku. Aplikácia navrhla využiť všetky modalít a skombinovať ich pomocou pravidla *sum-rule*. Na testovaných dátach bola dosiahnutá hodnota $EER = 0.0270767$ čo je mierne zlepšenie oproti pôvodnej hodnote. Na tréningových dátach bola dosiahnutá hodnota $EER = 0.026844$, čo je mierne zhoršenie. Dĺžka výpočtu sa zvýšila na 43 minút.

Automatický výber metódy fúzie – Aplikácia navrhla skombinovať modalít 0,2,3,4,5 pomocou *min-rule* pravidla. Na testovaných dátach bola dosiahnutá hodnota $EER = 0.109895$ čo je miernym zlepšením oproti predchádzajúcej hodnote. Na tréningových dátach bola dosiahnutá hodnota $EER = 0.123482$, čo je horší výsledok. Výpočet trval približne 26 minút.

Automatický výber metriky - Pri tomto experimente bolo na vstupe zadané pravidlo pre fúziu. Zvolil som pravidlo *min-rule*. Aplikácia navrhla použiť *city-block* metriku pre modalít 2,3, pre ostatné bola zvolená *korelačná* metrika. Kombinovali sa modalít 0,1,3,4. Na testovaných dátach bola dosiahnutá hodnota $EER = 0.0361817$, čo je výrazne zlepšenie a na tréningových dátach bola dosiahnutá hodnota $EER = 0.0334517$, čo je nižšia hodnota ako pôvodná. Dĺžka výpočtu sa zvýšila na 24 minút.

4.5 Fúzia na úrovni rozhodnutia

Automatický výber fúzie

Aplikácia navrhne ideálnu kombináciu pre fúziu rozhodnutí. Použitím tohto spôsobu kombinácie modalít a využitých techník by sme mali získať najlepšie výsledné FMR testovanej databázy.

Aplikácia navrhla skombinovať modalít 2,3,4,5. Pre jednotlivé modalít navrhla metriky s najnižším výsledným EER , a to *city-block* metriku pre modalít 2,3 a *korelačnú* metriku pre ostatné modalít. Pre fúziu bolo navrhnuté *AND* pravidlo. Výpočet trval necelé 4 minúty.

Touto kombináciou bolo na tréningových dátach dosiahnuté $FMR = 0.00283286$ a $FNMR = 0.245614$ a na testovacích dátach $FMR = 0.00865545$ a $FNMR = 0.22695$.

Automatický výber metódy fúzie

Pri tomto experimente boli zadane metriky pre jednotlivé modalities. Pre modalities 0,1 metrika *city-block*, pre modalities 2,3 *korelačná* metrika, pre modalities 4,5 *euklidovská* metrika. Kombinovali sa metriky 0,4,5 pomocou AND pravidla. Výpočet trval necelé 4 minúty.

Touto kombináciou bolo na tréningových dátach dosiahnuté $FMR = 0.0278565$ a $FNMR = 0.464912$ a na testovacích dátach $FMR = 0.026934$ a $FNMR = 0.409929$.

Automatický výber metriky

Pri tomto experimente bola zadná metóda fúzie. Vybral som pravidlo OR, keďže v predchádzajúcich prípadoch toto pravidlo nebolo použité. Pre jednotlivé modalities navrhla metriky s najnižším výsledným EER, a to *city-block* metriku pre modalities 2,3 a *korelačnú* metriku pre ostatné modalities. Kombinovali sa modalities 2,3,4,5. Výpočet trval 4 minúty.

Touto kombináciou bolo na tréningových dátach dosiahnuté $FMR = 0.290368$ a $FNMR = 0.00877193$ a na testovacích dátach $FMR = 0.247782$ a $FNMR = 0.00851064$.

Vplyv zväčšenia databáze na výsledok

Zvýšenie počtu modalít pri zachovaní počtu vektorov pri jednotlivých modalitách

Automatický výber fúzie – Na modalitách 0,1,2,3,6,7,9,10,11 bola použitá *korelačná* metrika a na modalitách 4,5,8 bola použitá *city-block* metrika. Výsledok bol kombinovaný z modalít 5,7,10,11. Bolo použité pravidlo AND. Na testovaných dátach bola dosiahnutá hodnota $FMR = 0.00263427$ a $FNMR = 0.248227$ čo je mierne zlepšenie oproti pôvodnej hodnote. Na tréningových dátach bola dosiahnutá hodnota $FMR = 0.00236072$ a $FNMR = 0.267544$, čo je lepšia hodnota. Výpočet trval 14 a pol minúty.

Automatický výber metódy fúzie – Pri tomto experimente boli zadane metriky pre jednotlivé modalities. Pre modalities 0,1,2,3 metrika *city-block*, pre modalities 4,5,6,7 *korelačná* metrika, pre modalities 8,9,10,11 *euklidovská* metrika. Kombinovali sa modalities 1,9,10,11. Použité pravidlo AND. Na testovaných dátach bola dosiahnutá hodnota $FMR = 0.0118811$ a $FNMR = 0.392908$. Na tréningových dátach bola dosiahnutá hodnota $FMR = 0.0174693$ a $FNMR = 0.416667$. Výpočet trval 4 minúty a 6 sekúnd.

Automatický výber metriky – Pre modalities 0,1,2,3,6,7,9,10,11 bola použitá *korelačná* metrika, pre zvyšné *city-block* metrika. Kombinovali sa modalities 5,7,10,11 pomocou zadaného OR pravidla. Na testovaných dátach bola dosiahnutá hodnota $FMR = 0.279985$ a $FNMR = 0.0070922$ čo je zlepšenie oproti pôvodnej hodnote. Na tréningových dátach bola dosiahnutá hodnota $FMR = 0.336638$ a $FNMR = 0.00877193$, čo je horšie ako pôvodná hodnota. Výpočet trval 5 minút a 20 sekúnd.

Zvýšenie počtu vektorov pre jednotlivé modalities

Počet vektorov pre tréovanie zvýšený na 160 a počet vektorov pre testovanie zvýšený na 380.

Automatický výber fúzie – Aplikácia navrhla *city-block* metriku pre modalities 2,3 a *korelačnú* metriku pre ostatné modalities. Aplikácia navrhla využiť modalities 2,3,4,5 a skombinovať ich pomocou pravidla *AND*. Na testovaných dátach bola dosiahnutá hodnota $FMR = 0.00371767$ a $FNMR = 0.281323$. Na tréovacích dátach bola dosiahnutá hodnota $FMR = 0.00330387$ a $FNMR = 0.251223$, čo je mierne zhoršenie. Dĺžka výpočtu sa zvýšila na 43 minút.

Automatický výber metódy fúzie – Pri tomto experimente boli zadane metriky pre jednotlivé modalities. Pre modalities 0,1 metrika *city-block*, pre modalities 2,3 *korelačná* metrika, pre modalities 4,5 *euklidovská* metrika. Kombinovali sa metriky 2,3,4,5 pomocou *AND* pravidla. Výpočet trval necelých 14 minút. Touto kombináciou bolo na tréovacích dátach dosiahnuté $FMR = 0.0124895$ a $FNMR = 0.420669$ a na testovacích dátach $FMR = 0.026934$ a $FNMR = 0.409929$.

Automatický výber metriky - Pri tomto experimente bola zadná metóda fúzie. Vybral som pravidlo *OR*. Aplikácia navrhla použiť *city-block* metriku pre modalities 2,3, pre ostatné bola zvolená *korelačná* metrika. Kombinovali sa modalities 0,1,3,4. Na testovaných dátach bola dosiahnutá hodnota $FMR = 0.261817$ a $FNMR = 0.010506$ a na tréovacích dátach bola dosiahnutá hodnota $FMR = 0.299331$ a $FNMR = 0.0130506$. Dĺžka výpočtu sa zvýšila na 24 minút.

4.6 Zhodnotenie

Pri menšej sade tréovacích dát, môže dôjsť k tomu, že sa vyberie kombinácia jednotlivých parametrov, ktorou dosiahneme najnižšie EER na tréovacích dátach ale na testovacích kvôli veľkému rozdielu už nie. Je to vidieť na automatickom výbere všetkých parametrov pri fúzii na úrovni vlastností, kde hodnota EER tréovacej časti bola veľmi nízka, ale naopak u testovacích dát bola vyššia ako pri inej kombinácii.

Zvýšením počtu modalít sme dosiahli zlepšenie výsledkov, keďže sme mali k dispozícii viac štatistických údajov.

Zvýšením počtu vektorov u jednotlivých modalít ale zásadne nezlepšilo výkon biometrického systému. Zásadne však ovplyvnilo rýchlosť výpočtu, keďže pridaním vektoru sa zvýši mnohonásobne počet možných kombinácií pri porovnaní.

5 Záver

Cieľom práce bol návrh, implementácia a testovanie aplikácie pre návrh postupu fúzie biometrických systémov do multi-biometrického systému a následná evaluáciu tohto systému. V prvej časti práce boli objasnené základné pojmy z biometrie. V ďalšej časti bol podrobnejšie popísaný návrh jednotlivých častí aplikácie a implementácia v jazyku c++.

Hlavným cieľom práce bolo vytvorenie funkčnej aplikácie, ktorá bude ovládaná cez príkazový riadok. Aplikácia dokáže navrhovať zo zadaných tréningových dát ideálny postup fúzie na úrovni skóre, rozhodnutia a vlastností.

Hlavným nedostatkom práce je nedostatočná optimalizácia, a preto výpočty na veľkých databázach trvajú aj viac ako hodinu. Ďalším nedostatkom je využitie len základných metód pre fúziu. Tieto nedostatky sa dajú ale odstrániť. Výpočet by sa dal odstrániť optimalizovaním, napríklad paralelne vypočítavať skóre pomocou rôznych metrík. Druhý nedostatok je ľahšie odstrániteľný, keďže aplikácia bola navrhnutá tak, aby nebol veľký problém pridávať ďalšie metódy.

Za prínos tejto práce považujem vytvorenie základu aplikácie, ktorú je možné ďalej rozvíjať. Osobným prínosom je získanie nových informácií o tejto problematike a získanie skúsenosti s programovacím jazykom c++ a knižnicou opencv.

Literatúra

- [1] Introduction to Biometrics. JAIN, Anil K, Patrick FLYNN a Arun A ROSS. *Handbook of biometrics*. New York: Springer, c2008, s. 1-23. ISBN 9780387710419.
- [2] Introduction to Multibiometrics. JAIN, Anil K, Patrick FLYNN a Arun A ROSS. *Handbook of biometrics*. New York: Springer, c2008, s. 271-293. ISBN 9780387710419.
- [3] RAK, Roman. *Biometrie a identita člověka ve forezních a komerčních aplikacích*. 1. vyd. Praha: Grada, 2008, s. 120-144. ISBN 978-80-247-2365-5.
- [4] Jolliffe, I. 2005. Principal Component Analysis. *Encyclopedia of Statistics in Behavioral Science*.
- [5] Table 14.1 Summary of score normalization techniques. In: JAIN, Anil K, Patrick FLYNN a Arun A ROSS. *Handbook of biometrics*. New York: Springer, c2008, s. 271-293. ISBN 9780387710419.

Seznam příloh

Příloha 1. Manuál ...

Příloha 2. Zdrojové texty ...

Příloha 3. CD/DVD ...